

Desenvolupament de llocs web amb PHP i MySQL

TEMA 2

TEORIA2

FORMULARIS. MANEIG DE FITXERS. PAS DE VARIABLES. SERVIDOR

2. [MANEIG DE FITXERS](#)

3. [MANEIG DE DIRECTORIS](#)

4. [EXECUCIÓ DE COMANDAMENTS DEL SISTEMA OPERATIU EN PHP](#)

6. [ENLLAÇOS D'INTERÉS](#)

2. MANEIG DE FITXERS.

Veure tots els exemples en:

ejemplos_archivosDirectorios.php

PHP disposa d'una sèrie de funcions que ens permeten manejar fitxers.

- **fopen(arxiu,mode) :**

Amb aquesta funció obrim un fitxer, bé siga local o una adreça d'internet (http:// o ftp://). Ens retorna un valor numèric (indicador d'arxiu) de tipus integer que ens servirà per a fer referència a l'arxiu obert. En cas de no poder obrir l'arxiu, retorna FALSE

Podem obrir un arxiu de les següents maneres:

- **r** sols lectura
- **r+** lectura i escriptura
- **w** sols escriptura. Si no existeix l'arxiu el crea, si ja existeix el matxaca.
- **w+** lectura i escriptura. Si no existeix l'arxiu el crea, si ja existeix el matxaca.
- **a** sols lectura. Si no existeix l'arxiu el crea, si ja existeix comença a escriure al final de l'arxiu.
- **a+** lectura i escriptura. Si no existeix l'arxiu el crea, si ja existeix comença a escriure al final de l'arxiu.
- Els modes **r**, **r+**, **w**, **w+** col·loquen el punter de lectura/escriptura a principi del fitxer,
- Els modes **a**, **a+** el col·loquen al final

exemple: fopen amb una URL:

```
<?php
//ejemplo de fopen
//abre un archivo utilizando el protocolo HTTP
$url="http://www.ciberaula.com/";
if (!fopen($url, "r"))
{
echo "El archivo no se puede abrir\n";
}
?>
```

- **feof(indicador_arxiu) :**

Aquesta funció retorna TRUE si el capçal de lectura/escriptura es troba al final del fitxer, i FALSE en cas contrari.

- **fgets(indicador_arxiu, [longitud]) :**

Ens retorna una cadena amb la *longitud* específica del fitxer al qual apunta l'indicador d'arxiu .

El paràmetre *longitud* és opcional. La lectura acaba quan s'han llegit *length - 1* bytes, o una nova línia (la qual està inclosa en el valor de retorn), o un *EOF* (el que succeeixca primer).

Si no s'especifica una longitud, es continuarà llegint des del flux fins que aconseguisca el final de la línia.

```
<?php
// Ejemplo de fgets
//abre un archivo e imprime cada linea
$nomArchivo="ARCHIVOS\\ej_gets.txt";
$fp = fopen( $nomArchivo, "r");
if ($fp)
{
    while (!feof($fp))
    {
        $linea = fgets($fp, 255);
        echo $linea;
        echo '<br>';
    }
    fclose ($fp);
}
?>
```

- **fgetc(resource \$handle) :**

Obté un caràcter d'un arxiu.

El fitxer al qual s'apunta ha de ser vàlid, i ha d'estar obert per fopen() .

Retorna una cadena que conté un **sol caràcter** llegit des de l'arxiu apuntat per handle.

Retorna FALSE sobre EOF.

Exemple1:

```
<?php
// Ejemplo de fgetc
$nomArchivo="ARCHIVOS\\ej_getc.txt";
$file = fopen($nomArchivo,"r");

while (! feof ($file))
{
    echo fgetc($file);
}
fclose($file);
?>
```

Exemple2:

Sobre l'entrada standard (és a dir, **teclat**)

```
<?php
echo 'Segur que desitja abandonar? (s/n) ';

$input = fgetc(STDIN);

if ($input == 's')

exit(0);}

?>
```

- **readfile(fitxer) :**

funció que visualitza el contingut d'un fitxer per pantalla

Exemple:

```
<?php
// Ejemplo Readfile
$nomArchivo="ARCHIVOS\\ej_readfile.txt";
if(file_exists($nomArchivo)) readfile($nomArchivo);
else
    echo "El archivo no existe";

?>
```

- **fwrite(indicador_arxiu, cadena, [length]):**

Escriu una cadena en el fitxer indicat, el qual ha d'haver sigut prèviament obert

La funció **fwrite** retorna el nombre de bytes escrits o FALSE, en cas que s'haja produït un error.

L'argument *length* és opcional. Si es dona, l'escriptura es detindrà després que s'aconsegueixca *length* nombre de bytes o que s'arribe al final del string, la qual cosa succeïska primer

```
<?php
// ejemplo fwrite
$nomArchivo="ARCHIVOS\\ej_fwrite.txt";
$fd1=fopen($nomArchivo,"w");
$res=fwrite($fd1,"Hola mundo!");
fclose($fd1);
?>
```

- **fputs(indicador_arxiu, cadena):**

Escriu una cadena en el fitxer indicat, el qual ha d'haver sigut prèviament obert.

La funció **fputs** retorna TRUE si s'ha escrit amb èxit, en cas contrari retorna FALSE.

```
<?php
// Ejemplo fputs
//abre un archivo y escribe en él
$nomArchivo="ARCHIVOS\\ej_fputs.txt";
$archivo = fopen($nomArchivo , "w");
if ($archivo)
{
    fputs ($archivo, "Hola Mundo");
}
fclose ($archivo);
?>
```


- **filesize(String \$filename)**

Retorna la grandària en bytes d'un arxiu.

Rep el nom del arxiu com a argument

De vegades es fa servir com a segon argument de la funció *fread*

```
<?php
//Ejemplo filesize
$nomArchivo="ARCHIVOS\\ej_filesize.txt";
$tamanyo_fichero= filesize($nomArchivo);
echo "<br>El tamaño del archivo <i><b>".$nomArchivo."</i></b> es:<br>";
echo "$tamanyo_fichero caracteres";
?>
```

- **fread(resource \$handle , int \$length)**

Llegeix fins a *length* bytes des del punter al fitxer referenciat per handle

La lectura acaba tan prompte com es trobe una de les següents condicions:

- *length* bytes han sigut llegits
- *EOF* (fi de fitxer) és aconseguit

Nota:

Normalment com a segon argument de *fread* es fa servir *filesize (\$nom_fitxer)*

Exemple:

```
<?php
// ejemplo fread
// poner el contenido de un fichero en una cadena
$nomArchivo="ARCHIVOS\\ej_fread.txt";
$gestor = fopen($nomArchivo, "r");
$contenido = fread($gestor, filesize($nomArchivo));
fclose($gestor);
?>
```

- **fclose (indicador_arxiu):**

Ens tanca el fitxer l'indicador del qual li diem .

Retorna TRUE si el fitxer es tanca correctament i FALSE si no s'ha pogut tancar.

- **file_exists (fitxer):**

Retorna TRUE si l'arxiu especificat existeix, i FALSE en cas contrari.

```
<?php
// ejemplo file_exists
$nomArchivo="ARCHIVOS\\ej_file_exists.txt";
if (file_exists($nomArchivo))
{
    echo "<br>El fichero existe";
}
else
{
    echo "<br>El fichero NO existe";
}
?>
```

- **copy (origen, destinació):**

Còpia un fitxer d'un lloc (origen) a un altre (destinació), retorna TRUE si la còpia ha tingut èxit i FALSE en cas contrari.

```
<?php
// ejemplo copy
$origen="ARCHIVOS\\ej_copyOrigen.txt";
$destino="ARCHIVOS\\ej_copyDestino.txt";
if (copy($origen, $destino))
{
    echo "<br>El fichero ha sido copiado con éxito";
}
else
{
    echo "<br>El fichero NO se ha podido copiar";
}
?>
```

- **rename(\$fich1,\$fich2) :**

Canvia el nom de l'arxiu de fich1 a fich2.

Exemple:

```
<?php
// ejemplo rename
$nomInicial="ARCHIVOS\\ej_renameInicial.txt";
$nomFinal="ARCHIVOS\\ej_renameFinal.txt";
rename($nomInicial,$nomFinal);
?>
```

- **unlink(\$arxiu) :**

Esborra l'arxiu

Exemple:

```
<?php
$nomArchivo="ARCHIVOS\\ej_unlink.txt";
unlink($nomArchivo);
?>
```

- **fseek (resource \$handle , int \$offset [, int \$whence = SEEK_SET]):**

Estableix l'indicador de posició de fitxer per al fitxer referenciat per handle

La nova posició, mesura en bytes des de l'inici del fitxer, s'obté afegint *offset* a la posició especificada per *whence*

Avança o retrocedeix el punter de l'arxiu un cert nombre de posicions

Si té èxit, retorna 0; en cas contrari retorna -1.

Veure enllaç:

<http://php.net/manual/es/function.fseek.php>

```
<?php
    $fp = fopen('fichero.txt','r');
    //leer alguna información
    $data= fgets($fp,4096);
    //volver al principio del fichero
    // igual que rewind($fp);
    fseek($fp,0);
?>
```

- **`fpassthru($id)` :**

Llegeix completament l'arxiu i el mostra.

En realitat *fpassthru* llegeix l'arxiu indicat des de la posició actual fins a EOF i escriu els resultats en el buffer d'eixida (*eixida standard*).

Actua sobre un arxiu que s'haja obert prèviament

Nota: Pot necessitar cridar a *rewind()* per a reiniciar el punter a l'arxiu al principi de l'arxiu si ja s'ha escrit informació en l'arxiu.

```
<?php
// ejemplo fpassthru
$nomArchivo="ARCHIVOS\\ej_fpassthru.txt";
$fp = fopen($nomArchivo, 'r');

// leemos la primera linea
$primLinea=fgets($fp);
echo "<br>Contenido de la primera línea:<br>";
echo $primLinea;

echo "<br>Contenido del resto del archivo:<br>";
fpassthru($fp);
?>
```

- **rewind (resource \$handle)**

Estableix l'indicador de posició d'arxiu de handle al principi del flux de l'arxiu.

Retorna TRUE en cas d'èxit. FALSE en cas contrari

```
<?php
    $nomArchivo="ARCHIVOS\\ej_rewind.txt";
    $fp = fopen($nomArchivo, 'r');
    // leemos la primera linea
    $primLinea=fgets($fp);
    echo "<br>Contenido de la primera línea:<br>";
    echo $primLinea;
    // leemos la segunda linea
    $segundaLinea=fgets($fp);
    echo "<br>Contenido de la segunda línea:<br>";
    echo $segundaLinea;
    // volver al principio del fichero
    rewind($fp);
    echo "Volvemos a leer de primera línea<br>";
    $Linea=fgets($fp);
    echo "<br>Contenido de la primera línea:<br>";
    echo $Linea;
?>
```


- **file(\$fitxer) :**

Bolca el contingut d'un fitxer en un array de tants elements com línies tinga el fitxer.

Exemple:

Passa el contingut de l'arxiu especificat a un array. Compta el nombre d'elements del array generat i mostra el seu contingut per pantalla.

Veure *ejemplo_fich2.php*:

```
<?php
    $arr=file('name.txt');
    // To check the number of lines
    echo count($arr).'\n';
    foreach($arr as $name)
    {
        echo $name.''\n';
    }
?>
```

Exemple completo1:

Veure *ejemplo_fich1.php*

Es comprova si existeix un arxiu. S'intenta obrir. En cas afirmatiu s'extrau el seu contingut. En un altre cas s'emet missatge. Finalment, es tanca

```
<?php
if(file_exists($nombre))
{
    if($ref=fopen($nombre,"r"))
    {
        while(!feof($ref))
        {
            $linea=fgets($ref,11);
            echo $linea;
        }
        fclose($ref);
    }
    else
        echo "Error al abrir archivo";
}
else
    echo "No existe archivo";
?>
```

Exemple complet 2:

Veure ejemplo _fich2.php

Es comprova si existeix el fitxer, en cas afirmatiu s'extrauen els elements d'un arxiu i es bolquen en un array amb funció **file**

En el bucle *while* extraiem un a un, en cada iteració, els elements del array *\$text* mitjançant la funció **each**.

Tot seguit, la funció **list** s'encarrega de separar, d'una banda, l'índex de l'element, el qual és guardat en la variable *\$línia*, i, per un altre, el contingut de l'element, que és guardat en la variable *\$contingut*. El contingut de totes dues variables, *\$línia* i *\$contingut*, són mostrats en el navegador.

```
<?php
    $nombre="prueba.txt";
    if(file_exists($nombre))
    {
        // Con "file" creamos array "texto".
        //Cada línea del fich será un elem del array
        $texto=file($nombre);
        // Mostramos cada una de las líneas del array 'list' y 'each'
        while(list($línea,$contenido)=each($texto))
        {
            echo ($línea+1). " --> ".$contenido."<br>";
        }
    }
    else
        echo "El archivo no existe";
?>
```

3. MANEIG DE DIRECTORIS.

Veurem algunes funcions que ens poden ser molt útils en la navegació per directoris i fins i tot per a crear exploradors d'arxius en el nostre navegador.

Podem distingir tres accions elementals:

- Obertura del directori, assignant-li un identificador.
- Realització de les tasques necessàries en relació amb aqueix directori
- Tancament del directori

FUNCIONS PER A GESTIÓ D'ARXIUS/CARPETES DEL SISTEMA OPERATIU:

Veure arxiu: ***ejemplos_comandos_directorios.php***

Funciones de gestión de directorios		
Función	Descripción	Sintaxis
opendir	Abre un directorio situado en \$path y le asigna un identificador \$dir	\$dir = opendir (\$path)
readdir	Lee un elemento del directorio \$dir abierto previamente con opendir y desplaza el puntero al elemento siguiente	readdir (\$dir)
rmdir	Elimina el directorio \$dir	rmdir(\$dir)
mkdir	Crea un directorio situado en \$path con los derechos de acceso \$derechos (entero)	mkdir (\$path, \$derechos)
rewinddir	Vuelve el puntero de lectura del directorio \$dir al primer elemento	rewinddir (\$dir)
closedir	Cierra el directorio \$dir abierto previamente con opendir	closedir (\$dir)

https://www.tutorialspoint.com/php/php_directory_functions.htm

- **resource opendir (string \$path [, resource \$context]);**

Intenta obrir la carpeta que se li indica en *\$path*

Obri un gestor de directori per a ser usat en crides posteriors com *closedir()*, *readdir()*, i *rewinddir()*.

El paràmetre *\$context* no és obligatori i indica diverses opcions que poden modificar el comportament.

Exemple:

```
<?php
$ruta='D:\\projects_PHP_IAW\\UPLOADS2/';
$des= @opendir($ruta); // incluimos @ para evitar que
                        // aparezca error cuando no existe
if ($des==true) {
    echo "Existe la carpeta";
}
else
{
    echo "No Existe";
}
?>
```

- **String readdir ([resource \$dir_handle]);**

Retorna el nom de la següent entrada del directori. Les entrades són retornades en l'ordre en què van ser emmagatzemades pel sistema de fitxers.

Retorna el nom de l'entrada en cas d'èxit o FALSE en cas d'error.

El gestor de directori resource prèviament obert per *opendir()* . Si el gestor de directori no s'especifica, la última connexió oberta per *opendir()* és assumida.

Exemple:

```
//readdir()  
$ruta='D:\\projects_PHP_IAW\\UPLOADS';  
$gestor = opendir($ruta);  
if ($gestor) {  
    echo "Identificador de gestor de directorio: <b>".$gestor."</b><br>";  
    echo "<b>Contenido de la carpeta: </b><br>";  
  
    // Esta es la forma correcta de iterar sobre el directorio.  
    $entrada = readdir($gestor);  
    while ($entrada !== false )  
    {  
        echo $entrada;  
        echo '<br>';  
        $entrada = readdir($gestor);  
    }  
    closedir($gestor);  
}
```

- **array** scandir (**string** \$directory [, **int** \$sorting_order = SCANDIR_SORT_ASCENDING [, **resource** \$context]])

Rep com a argument **\$directory**, la ruta i la forma d'ordenació,

Enumera els fitxers i directoris situats en la ruta especificada

Retorna un array **associatiu** en el qual en cada posició està un string amb el nom de cada arxiu.

```
//scandir()
$ruta='D:\\projects_PHP_IAW\\UPLOADS';
$fitxeros1 = scandir($ruta); // sólo visualiza los nombres de archivo
$fitxeros2 = scandir($ruta , 1); // tiene en cuenta los ficheros . y ..

print_r($fitxeros1);
echo '<br>';
print_r($fitxeros2);
echo '<br>';

// los podemos recorrer con un foreach
foreach ($fitxeros1 as $key => $value)
{
echo $key.'====>'.$value.'<br>';
}
```


- **bool chdir (string \$directory)**

Canvia de directori

Rep com a argument *\$directory* la ruta

```
//chdir()  
$ruta='D:\\projects_PHP_IAW\\UPLOADS';  
  
// directorio actual  
$dirActual=getcwd();  
echo "El directorio actual es: <b>".$dirActual."</b><br>";  
  
echo "Cambiamos de directorio....: <br>";  
chdir($ruta);  
  
// directorio actual  
$dirNuevo=getcwd();  
echo "El nuevo directorio es: <b>".$dirNuevo."</b><br>";
```

- **bool chroot (string \$directory)**

Canvia el directori arrel del procés actual a directori, i canvia el directori de treball actual a "/.

Aquesta funció només es troba disponible a sistemes GNU i BSD, i es troba usant un entorn CLI, CGI o SAPI embegut.

Així mateix, aquesta funció requereix privilegis d'administrador (root).

```
//rchroot()  
chroot("/ruta/al/chroot/");
```

- `bool mkdir (string $pathname [, int $mode = 0777 [, bool $recursive = false [, resource $context]]])`

Crea un directori.

Retorna TRUE en cas d'èxit i FALSE en cas d'error

Arguments, només el nom de la carpeta a crear és obligatori, és resta són opcionals

pathname: La ruta del directori.

mode: El mode predeterminat és 0777, cosa que significa l'accés més ampli possible.

Per a més informació sobre els modes, llija els detalls en la pàgina de *chmod()*.

Nota: mode és ignorat en Windows.

Observe que probablement es vol especificar la manera com un número octal, cosa que significa que hauria d'haver-hi un zero inicial. El mode és modificat també per l'actual màscara d'usuari, la qual es pot canviar usant *umask()*.

Recursive: permet la creació de directoris niats especificat en el paràmetre *pathname*.

Exemple:

```
//mkdir()
$ruta='D:\\projects_PHP_IAW\\UPLOADS\\NIVEL1';
echo "Creamos la carpeta<br>";
mkdir($ruta, 0777, true);
echo "Asignamos permisos<br>";
chmod($ruta, 0777);
```

- `bool rmdir (string $dirname [, resource $context])`

Esborra el directori especificat

Retorna TRUE en cas d'èxit i FALSE en cas d'error

```
//rmdir()
$ruta='D:\\projects_PHP_IAW\\UPLOADS\\NIVEL1';
echo "Vamos a borrar la carpeta<br>";

if (!is_dir('NIVEL1'))
{
    echo "Se trata de una carpeta.<br>";
    $res=rmdir($ruta);
    if ($res==true)
        echo "Carpeta Borrada<br>";
    else
        echo "No se ha borrado la carpeta<br>";
}
```

- `void rewinddir ([resource $dir_handle])`

Tornar el gestor de directori

```
//rewinddir()
$ruta='D:\\projects_PHP_IAW\\UPLOADS';
```

- **bool is_dir (string \$filename)**

Retorna TRUE si *\$filename* és un directori i FALSE en cas contrari

- **bool is_file (string \$filename)**

Retorna TRUE si *\$filename* és un arxiu i FALSE en cas contrari

- **void closedir ([resource \$dir_handle])**

Tanca el directori que s'indica amb el se handler

El gestor de directori tipus *resource*, prèviament obert amb *opendir()*. Si el gestor de directori no s'especifica, l'última connexió oberta per *opendir()* és assumida.

```
$ruta='D:\\projects_PHP_IAW\\UPLOADS';
$gestor = opendir($ruta);

// tratamiento del directorio
// cierre
closedir($gestor);
```

Nota 1:

Perquè un directori pugui ser esborrat ha de estar buit

Nota 2:

`disk_total_space()`, `disk_free_space()`: podem obtenir l'espai total i lliure, en bytes, que correspon a un cert directori, la ruta del qual indicarem com a argument.

Nota 3:

La constant `_FILE_` conté el camí complet i el nom de l'arxiu on està contingut el codi que està executant-se.

Funcions útils per a maneig de fitxers:

- **dirname()** → pren com a argument una ruta completa d'un fitxer i retorna la part corresponent al directori.
- **basename()** → pren com a argument una ruta completa d'un fitxer i retorna nom i extensió de l'arxiu .

Exemple:

```
<?php
// $c contiene el número de bytes disponibles en C
$C=disk_free_space("C:");
$T=disk_total_space("C:");
echo "Bytes libres en la unidad c: ".$C."<br>";
echo "Bytes totales en la unidad c: ".$T."<br>";
?>
```

Exemple:

```
<?php
$directorio = opendir("../..//Imagenes/");
while ($fichero=readdir($directorio))
{
    if (is_dir($fichero))
        print "<b>$fichero</b> es un directorio<br>";
    else
        print "<b>$fichero</b> es un fichero<br>";
}
?>
```

4. EXECUCIÓ DE COMANDOS DEL SISTEMA OPERATIU EN PHP

FORMA 1: FITXERS .PHP QUE CONTENEN COMANDAMENTS LINUX

Des d'un programa en PHP es pot invocar a comandaments del sistema operatiu.

Es tracta de fitxers .php que contenen comandaments linux

Veure manual en:

<http://php.net/manual/es/book.exec.php>

Es pot fer de diverses formes:

- exec
- system
- shell_exec
- cometes invertides

Les veiem en la següent imatge i posteriorment es detalla cadascuna d'elles:

```
<?php
    $cmd = "whoami";
    echo ` $cmd `;
    echo shell_exec($cmd);
    system($cmd);
?>
```

- `string exec (string $command [, array &$output [, int &$return_var]])`

Executa el comando donat.

Paràmetres

command

El comando que serà executat.

output

Si l'argument **output** és present, llavors el array especificat serà omplit amb cada línia de l'eixida del comando. L'espai en blanc extra, com a `\n`, no és inclòs en aquest array. Note que si el array ja conté alguns elements, **exec()** annexarà els seus resultats al final del array. Si no desitja que la funció annexe els elements, use [unset\(\)](#) sobre el array abans de passar-la a `exec()`.

return_var

Si l'argument **return_var** és present juntament amb l'argument `output`, llavors l'estatus de retorn del comando executat serà escrit en aquesta variable.

Exemple:

```
$ruta='D:\\projects_PHP_IAW\\UPLOADS/';  
chdir($ruta); // Cambiaría al directorio que quiero visualizar  
  
// versión exec  
echo "<strong>versión mediante exec</strong>";  
echo '<pre>';  
exec ('dir',$result);  
//en el vector se se indica en el segundo argumento.  
//Por ello habrá que recorrer dicho vector  
  
foreach($result as $line)  
    echo $line."<br>";  
echo '</pre>';
```

- `string system (string $command [, int &$return_var])`

Executa un programa extern i mostra la seua eixida

Paràmetres

comando

El comando que serà executat.

return_var

Si l'argument **return_var** es troba present, llavors l'estatus retornat pel comando executat serà emmagatzemat en aquesta variable.

Valors retornats

Retorna l'última línia de l'eixida del comando en cas de tindre èxit, i **FALSE** si ocorre un error.

Exemple:

```
// versión system

//Vuelca la salida línea a línea a la pantalla
//del navegador
echo "<strong>Versión mediante system</strong>";
echo '<pre>';
$result=system('dir');
echo '</pre>';
```

- **string** shell_exec (**string** \$cmd)

Paràmetres

cmd: El comando que serà executat.

Valors retornats

L'eixida del comando executat o NULL si ocorre un error o el comando no produeix cap eixida.

```
<?php
$salida = shell_exec('ls -lart');
echo "<pre>$salida</pre>";
?>
```

- **Operador de cometa invertida**

PHP suporta un operador d'execució: les cometes invertides (` `).

PHP intentarà executar el contingut entre les cometes invertides com si es tractara d'un comando del shell;

L'eixida serà retornada (és a dir, no serà simplement bolcada com a eixida; pot ser assignada a una variable).

L'ús de l'operador de cometes invertides és idèntic al de *shell_exec()*.

Exemple:

```
// versión con comillas invertidas
echo "<strong>Versión con comillas invertidas</strong>";
echo '<pre>';
$result=`dir`;
echo $result;
echo '</pre>';
```

FORMA 2: FITXERS .PHP QUE CRIDEN A ARXIU .SH QUE CONTÉ COMANDAMENTS LINUX

Ara tenim dos arxius:

- un script linux (**shell script**)
- un **programa php**. Aquest és l'encarregat d'invocar al **shell script**

comandosLin0.php:

```
<?php
    $comando = shell_exec('sh comandosLin0.sh');
    echo "<pre>$comando</pre>";
?>
```

comandosLin0.sh:

```
#####
#!/bin/sh
#####

echo 'LISTADO CARPETA ACTUAL:<br>';
ls
```

Forma d'invocació:

```
// Forma1: Con shell_exec
    $comando=shell_exec('sh comandosLin0.sh');
    echo "<pre>$comando</pre>";
```

6. ENLLAÇOS D'INTERÉS

Tutorial HTML:

<http://www.htmlquick.com/es/tutorials.html>

Formularis:

<http://www.htmlquick.com/es/tutorials/forms/4.html#practice>